Algorithms for Data Science: The EM Algorithm

Héctor Corrada Bravo

University of Maryland, College Park, USA

DATA 606: 2020-04-28

Soft K-means Clustering

Instead of the combinatorial approach of the K-means algorithm, take a more direct probabilistic approach to modeling distribution $P(X)$.

Assume each of the K clusters corresponds to a multivariate distribution $P_k(X)$,

$P(X)$ is then a *mixture* of these distributions as $P(X) = \sum_{k=1}^{K} \pi_k P_k(X)$.

Soft K-means Clustering

Specifically, take $P_k(X)$ as a multivariate normal distribution $f_k(X) = N(\mu_k, \sigma_k^2 I)$

and mixture density $f(X) = \sum_{k=1}^{K} \pi_k f_k(X)$.

Soft K-means Clustering

Use Maximum Likelihood to estimate parameters

$$\theta = (\mu_1, \ldots, \mu_K, \sigma_1^2, \ldots, \sigma_K^2, \pi_1, \ldots, \pi_K)$$

based on their log-likelihood

$$\ell(\theta; X) = \sum_{i=1}^{N} \log \left[\sum_{k=1}^{K} \pi_k f_k(x_i; \theta) \right]$$

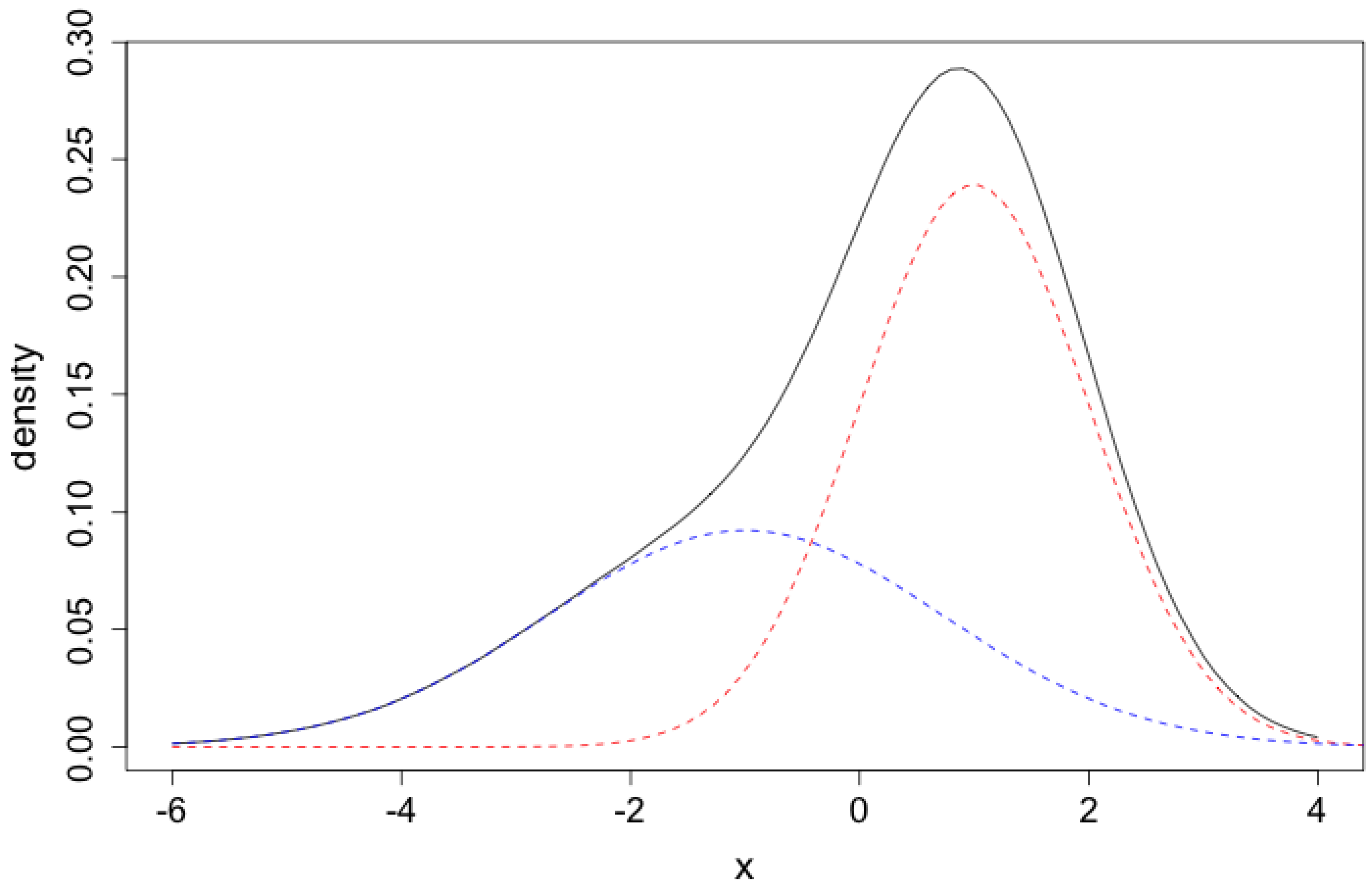
Soft K-means Clustering

$$\ell(\theta; X) = \sum_{i=1}^{N} \log\left[\sum_{k=1}^{K} \pi_k f_k(x_i; \theta)\right]$$

Maximizing this likelihood directly is computationally difficult

Use Expectation Maximization algorithm (EM) instead.

Example: Mixture of Two Univariate Gaussians



Soft K-means Clustering

Consider unobserved latent variables Δ_{ik} taking values 0 or 1,

$\Delta_{ij} = 1$ specifies observation x_i was generated by component k of the mixture distribution.

Soft K-means Clustering

Now set $Pr(\Delta_{ik} = 1) = \pi_k$,and assume we *observed* values for latent variables Δ_{ik}.

We can write the log-likelihood in this case as

$$\ell_0(\theta; X, \Delta) = \sum_{i=1}^{N} \sum_{k=1}^{K} \Delta_{ik} \log f_k(x_i; \theta) + \sum_{i=1}^{N} \sum_{k=1}^{K} \Delta_{ik} \log \pi_k$$

Soft K-means Clustering

We have closed-form solutions for maximum likelihood estimates:

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} \Delta_{ik} x_i}{\sum_{i=1}^{N} \Delta_{ik}}$$

$$\hat{\sigma}_k^2 = \frac{\sum_{i=1}^{N} \Delta_{ik} (x_i - \hat{\mu}_k)^2}{\sum_{i=1}^{N} \Delta_{ik}}$$

$$\hat{\pi}_k = \frac{\sum_{i=1}^{K} \Delta_{ik}}{N}.$$

Constrained optimization

We have a problem of type

$$\begin{array}{rc} \min_x & f_0(x) \\ \text{s.t.} & f_i(x) \leq 0 \; i = 1, \ldots, m \\ & h_i(x) = 0 \; i = 1, \ldots, p \end{array}$$

Note: This discussion follows Boyd and Vandenberghe, *Convex Optimization*

Constrained optimization

To solve these type of problems we will look at the *Lagrangian* function:

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i g_i(x)$$

Constrained optimization

There is a beautiful result giving *optimality conditions* based on the Lagrangian:

Suppose $\tilde{x}$, $\tilde{\lambda}$ and $\tilde{\nu}$ are *optimal*, then

$$f_i(\tilde{x}) \leq 0$$
$$h_i(\tilde{x}) = 0$$
$$\tilde{\lambda}_i \geq 0$$
$$\tilde{\lambda}_i f_i(\tilde{x}) = 0$$
$$\nabla L(\tilde{x}, \tilde{\lambda}, \tilde{\nu}) = 0$$

Constrained optimization

We can use the gradient and feasibility conditions to prove the MLE result.

Soft K-means Clustering

Of course, this result depends on observing values for Δ_{ik} which *we don't observe*. Use an iterative approach as well:

- given current estimate of parameters θ,
- Substitute $E[\Delta_{ik}|X_i, \theta]$ for Δ_{ik}.

Soft K-means Clustering

Of course, this result depends on observing values for Δ_{ik} which *we don't observe*. Use an iterative approach as well:

- given current estimate of parameters θ,
- Substitute $E[\Delta_{ik}|X_i, \theta]$ for Δ_{ik}.

We will prove that this maximizes the likelihood we need $\ell(\theta; X)$.

Soft K-means Clustering

Soft K-means Clustering

In the mixture case, what does this look like?

Define

$$\gamma_{ik}(\theta) = E(\Delta_{ik}|X_i, \theta) = Pr(\Delta_{ik} = 1|X_i, \theta)$$

Soft K-means Clustering

Soft K-means Clustering

Use Bayes' Rule to write this in terms of the multivariate normal densities with respect to current estimates θ:

$$\begin{aligned} \gamma_{ik} &= \frac{Pr(X_i|\Delta_{ik}=1)Pr(\Delta_{ik}=1)}{Pr(X_i)} \\ &= \frac{f_k(x_i;\mu_k,\sigma_k^2)\pi_k}{\sum_{l=1}^K f_l(x_i;\mu_l,\sigma_l^2)\pi_l} \end{aligned}$$

Soft K-means Clustering

Soft K-means Clustering

Quantity $\gamma_{ik}(\theta)$ is referred to as the *responsibility* of cluster k for observation i, according to current parameter estimate θ.

Soft K-means Clustering

Soft K-means Clustering

We can now give a complete specification of the EM algorithm for mixture model clustering.

1. Take initial guesses for parameters θ
2. *Expectation Step*: Compute responsibilities $\gamma_{ik}(\theta)$
3. *Maximization Step*: Estimate new parameters based on responsibilities as below.
4. Iterate steps 2 and 3 until convergence

Soft K-means Clustering

Soft K-means Algorithm

Estimates in the Maximization step are given by

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} \gamma_{ik}(\theta) x_i}{\sum_{i=1}^{N} \gamma_{ik}}$$

$$\hat{\sigma}_k^2 = \frac{\sum_{i=1}^{N} \gamma_{ik}(\theta)(x_i - \mu_k)^2}{\sum_{i=1}^{N} \gamma_{ik}(\theta)}$$

and

Soft K-means Clustering

Soft K-means Algorithm

The name "soft" K-means refers to the fact that parameter estimates for each cluster are obtained by weighted averages across all observations.

The EM Algorithm in General

So, why does that work?

Why does plugging in $\gamma_{ik}(\theta)$ for the latent variables Δ_{ik} work?

Why does that maximize log-likelihood $\ell(\theta; X)$?

The EM Algorithm in General

Think of it as follows:

Z: observed data
Z^m: missing *latent* data $T = (Z, Z^m)$: complete data (observed and missing)

The EM Algorithm in General

Think of it as follows:

Z: observed data
Z^m: missing *latent* data $T = (Z, Z^m)$: complete data (observed and missing)

$\ell(\theta'; Z)$: log-likehood w.r.t. *observed* data
$\ell_0(\theta'; T)$: log-likelihood w.r.t. *complete* data

The EM Algorithm in General

Next, notice that

$$Pr(Z|\theta') = \frac{Pr(T|\theta')}{Pr(Z^m|Z,\theta')}$$

The EM Algorithm in General

Next, notice that

$$Pr(Z|\theta') = \frac{Pr(T|\theta')}{Pr(Z^m|Z, \theta')}$$

As likelihood:

$$\ell(\theta'; Z) = \ell_0(\theta'; T) - \ell_1(\theta'; Z^m|Z)$$

The EM Algorithm in General

Iterative approach: given parameters θ take expectation of log-likelihoods

$$\begin{aligned}\ell(\theta';Z) &= E[\ell_0(\theta';T)|Z,\theta] - E[\ell_1(\theta';Z^m|Z)|Z,\theta] \\ &\equiv Q(\theta',\theta) - R(\theta',\theta)\end{aligned}$$

The EM Algorithm in General

Iterative approach: given parameters θ take expectation of log-likelihoods

$$\begin{aligned} \ell(\theta'; Z) &= E[\ell_0(\theta'; T)|Z, \theta] - E[\ell_1(\theta'; Z^m|Z)|Z, \theta] \\ &\equiv Q(\theta', \theta) - R(\theta', \theta) \end{aligned}$$

In soft k-means, $Q(\theta', \theta)$ is the log likelihood of complete data with Δ_{ik} replaced by $\gamma_{ik}(\theta)$

The EM Algorithm in General

The general EM algorithm

1. Initialize parameters $\theta^{(0)}$
2. Construct *function* $Q(\theta', \theta^{(j)})$
3. Find next set of parameters $\theta^{(j+1)} = \arg\max_{\theta'} Q(\theta', \theta^{(j)})$
4. Iterate steps 2 and 3 until convergence

The EM Algorithm in General

So, why does that work?

$$
\begin{aligned}
\ell(\theta^{(j+1)}; Z) - \ell(\theta^{(j)}; Z) = & \quad [Q(\theta^{(j+1)}, \theta^{(j)}) - Q(\theta^{(j)}, \theta^{(j)})] \\
& -[R(\theta^{(j+1)}, \theta^{(j)}) - R(\theta^{(j)}, \theta^{(j)})] \\
\geq & \quad 0
\end{aligned}
$$

The EM Algorithm in General

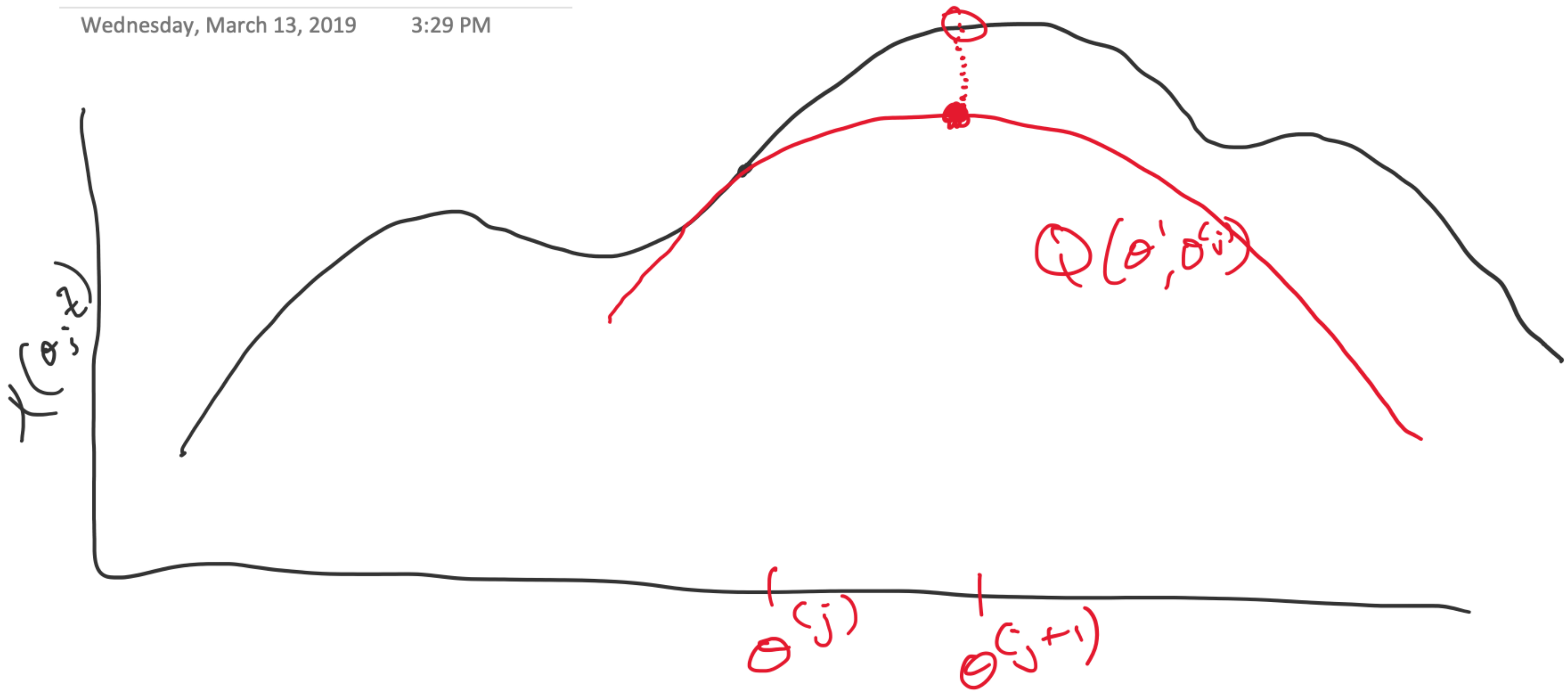
So, why does that work?

$$
\begin{aligned}
\ell(\theta^{(j+1)}; Z) - \ell(\theta^{(j)}; Z) = & \quad [Q(\theta^{(j+1)}, \theta^{(j)}) - Q(\theta^{(j)}, \theta^{(j)})] \\
& -[R(\theta^{(j+1)}, \theta^{(j)}) - R(\theta^{(j)}, \theta^{(j)})] \\
\geq & \quad 0
\end{aligned}
$$

I.E., every step makes log-likehood larger

The EM Algorithm in General

Why else does it work? $Q(\theta', \theta)$ *minorizes* $\ell(\theta'; Z)$



The EM Algorithm in General

General algorithmic concept:

Iterative approach:

- Initialize parameters
- Construct bound based on current parameters
- Optimize bound

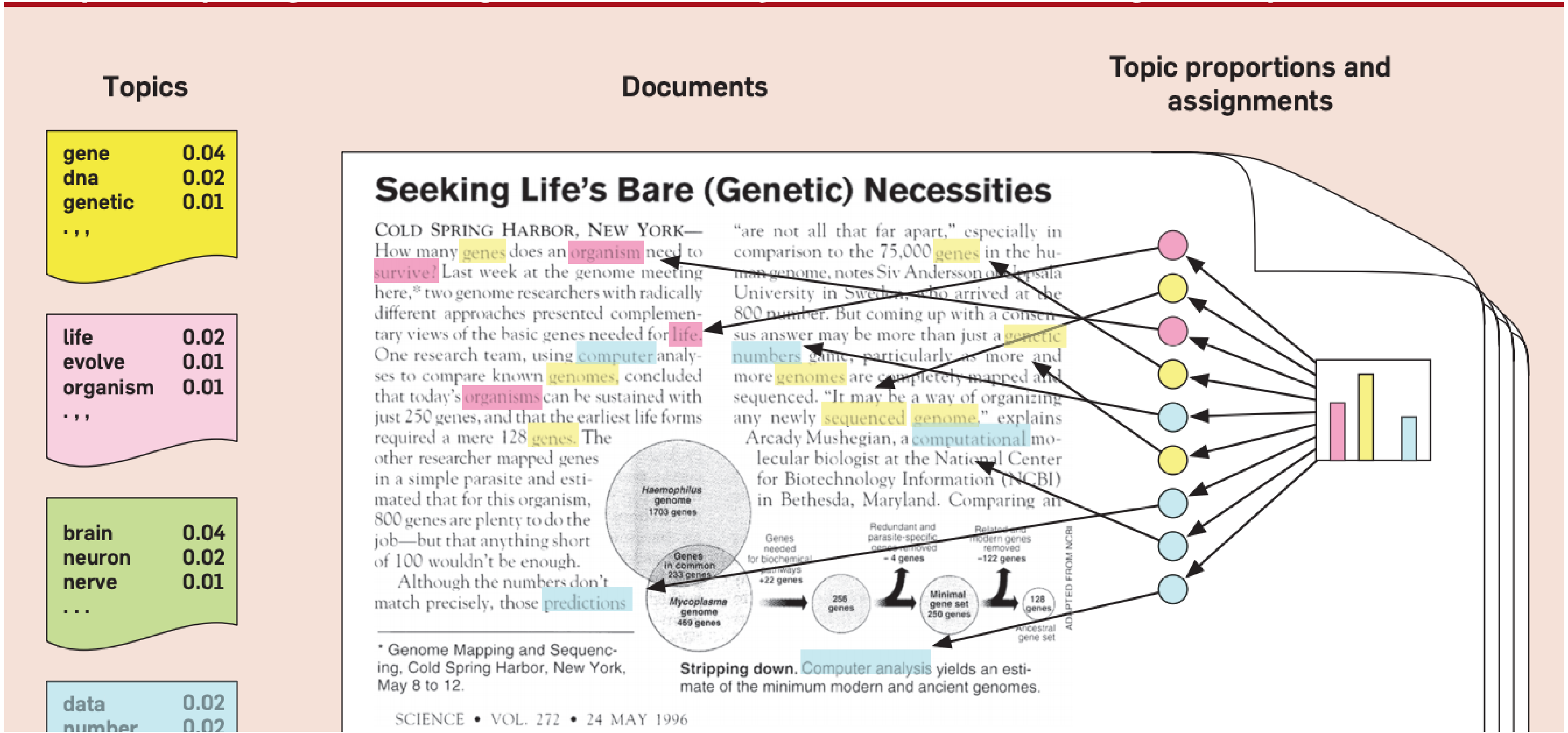
Imputing missing data

Z: observed data
Z^m: missing observations

Requires a likelihood model...

Latent semantic analysis

Documents as *mixtures* of topics (Hoffman 1998)



Latent semantic analysis

We have a set of documents D

Each document modeled as a bag-of-words (bow) over dictionary W.

$x_{w,d}$: the number of times word $w \in W$ appears in document $d \in D$.

Latent semantic analysis

Let's start with a simple model based on the frequency of word occurrences.

Each document is modeled as n_d draws from a *Multinomial* distribution with parameters $\theta_d = \{\theta_{1,d}, \ldots, \theta_{W,d}\}$

Note $\theta_{w,d} \geq 0$ and $\sum_w \theta_{w,d} = 1$.

Latent semantic analysis

Probability of observed corpus D

$$Pr(D|\{\theta_d\}) \propto \prod_{d=1}^{D} \prod_{w=1}^{W} \theta_{w,d}^{x_{w,d}}$$

Latent semantic analysis

Problem 1:

Prove MLE $\hat{\theta}_{w,d} = \frac{x_{w,d}}{n_d}$

Probablistic Latent Semantic Analysis

Let's change our document model to introduce topics.

The key idea is that the probability of observing a *word* in a *document* is given by two pieces:

- The probability of observing a *topic* in a document, and
- The probability of observing a *word* given a *topic*

$$Pr(w,d) = \sum_{t=1}^{T} Pr(w|t)Pr(t|d)$$

Probablistic Latent Semantic Analysis

So, we rewrite corpus probability as

$$Pr(D|\{p_d\}\{\theta_t\}) \propto \prod_{d=1}^{D} \prod_{w=1}^{W} \left(\sum_{t=1}^{T} p_{t,d}\theta_{w,t} \right)^{x_{w,d}}$$

Probablistic Latent Semantic Analysis

So, we rewrite corpus probability as

$$Pr(D|\{p_d\}\{\theta_t\}) \propto \prod_{d=1}^{D} \prod_{w=1}^{W} \left(\sum_{t=1}^{T} p_{t,d}\theta_{w,t} \right)^{x_{w,d}}$$

Mixture of topics!!

Probablistic Latent Semantic Analysis

A fully observed model

Assume you know the *latent* number of occurences of word w in document d generated from topic t:

$\Delta_{w,d,t}$, such that $\sum_t \Delta_{w,d,t} = x_{w,d}$.

In that case we can rewrite corpus probability:

$$Pr(D|\{p_d\},\{\theta_t\}) \propto \prod_{d=1}^{D} \prod_{w=1}^{W} \prod_{t=1}^{T} (p_{t,d}\theta_{w,t})^{\Delta_{w,d,t}}$$

Probablistic Latent Semantic Analysis

Problem 2 Show MLEs given by

$$\hat{p}_{t,d} = \frac{\sum_{w=1}^{W} \Delta_{w,d,t}}{\sum_{t=1}^{T} \sum_{w=1}^{W} \Delta_{w,d,t}}$$

$$\hat{\theta}_{w,t} = \frac{\sum_{d=1}^{D} \Delta_{w,d,t}}{\sum_{w=1}^{W} \sum_{d=1}^{D} \Delta_{w,d,t}}$$

Probablistic Latent Semantic Analysis

Since we don't observe $\Delta_{w,d,t}$ we use the EM algorithm

At each iteration (given current parameters $\{p_d\}$ and $\{\theta_d\}$ find *responsibility*

$$\gamma_{w,d,t} = E[\Delta_{w,d,t} | \{p_d\}, \{\theta_t\}]$$

and maximize fully observed likelihood plugging in $\gamma_{w,d,t}$ for $\Delta_{w,d,t}$

Probablistic Latent Semantic Analysis

Problem 4: Show

$$\gamma_{w,d,t} = x_{w,d} \times \frac{p_{t,d}\theta_{w,t}}{\sum_{t'=1}^{T} p_{t',d}\theta_{w,t'}}$$